ETHAN CHU

ethanchu@cs.cmu.edu | https://gediminas19.github.io/

EDUCATION

Carnegie Mellon University, Ph.D. in Computer Science | Advisor: Jan Hoffmann
Pittsburgh, PA | *Aug 2024 – Present*

Carnegie Mellon University, M.S. in Computer Science with Research Thesis
Pittsburgh, PA | *Jan 2023 – Dec 2023* GPA: 4.04/4.33

Carnegie Mellon University, B.S. in Computer Science with Systems Concentration
Pittsburgh, PA | *Aug 2019 – Dec 2022* GPA: 3.92/4.00 University Honors

TECHNICAL SKILLS

Fluent
C • C++ • Python • Rust • OCaml • SML • LaTeX

Competent
x86 asm • LLVM • Go • Ruby (on Rails) • JavaScript • Haskell • Agda

RESEARCH EXPERIENCE

Automation of Amortized Resource Analysis (AARA) via Resource-Aware ML 2 (RaML2) | Professor Jan Hoffmann | OCaml
Pittsburgh, PA | *Aug 2022 – Present*
RaML

- Designed and implemented new Intermediate Representations (IRs), typecheckers, and cost-aware interpreter for RaML 2
- Engineered compiler passes to parse and transform SML code into the IRs, involving pattern flattening and defunctionalization
- Augmented and implemented a new resource analysis type system that can analyze programs containing arbitrary recursive types
- Wrote my MS research thesis on my implementation and results, published in the CMU SCS Technical Report Collection
- Extended RaML 2 to support handling effects and exceptions and working to publish results at POPL (arXiv link forthcoming)
- Developed a novel "backflow" type system that achieves even tighter resource bounds and working toward presenting results

Verification of AARA in DeCalf | Professors Jan Hoffmann + Bob Harper | Agda
Pittsburgh, PA | *Aug 2024 – Present*

- Designed a Call-by-Push-Value (CBPV) based language (Giralf) with the cost effect and accompanying substructural type system
- Collaborated with authors of DeCalf to develop a categorical semantics encoding the soundness of AARA on Giralf into DeCalf
- Implemented in Agda a verified compiler from Giralf into a DeCalf proof of a resource bound on the original Giralf program
- Added an interface with RaML2, thus supporting a complete pipeline that infers then verifies resource bounds on SML programs
- Currently supports linear resource bounds, working toward polynomial resource bounds

Optimizing Compilers (15-795) Course Project: Incremental Function Inlining | Professor Todd Mowry | C++
Pittsburgh, PA | *Mar 2022 – May 2022*

- Adapted novel compiler algorithm for incremental (function) inlining into the V8 JavaScript compiler, achieving modest speedups
- Wrote a paper and gave a poster presentation discussing the implementation and results

Compiler Design (15-411) Course Project: Optimizations and Functional Programming Features | Rust
Pittsburgh, PA | *Nov 2021 – Dec 2021*

- Studied and implemented compiler optimizations including Partial Redundancy Elimination (PRE) and Aggressive Dead Code Elimination (ADCE) for a C to x86 asm compiler for Compiler Design course (15411), achieving speedups on par with GCC -O1
- Expanded compiler to support algebraic datatypes (ADTs) and function closures, including parsing, typechecking, and codegen

WORK EXPERIENCE

Galois | Static Analysis Intern | Haskell, C++
Portland, OR | *Jun 2023 – Aug 2023*

- Implemented in Haskell a static analysis pass over C++ code collecting scope information of every identifier (variables, classes, etc.)
- Built and documented a test suite ensuring that a One Definition Rule (ODR) static analysis pass fully covered each paragraph in the ODR section of the official C++ standard

Apple | Swift Foundation Framework Team Intern | Swift, C++, LLVM
Santa Clara, CA | *May 2022 – Aug 2022*

- Solved a confidential open problem regarding Swift Key Paths and implemented my solution into the Swift Compiler and Runtime
- Presented and demonstrated my new features in action to the head of the Swift Platform Experience team

Hudson River Trading | Systems Engineer Intern | SaltStack, Python, C
New York, NY | *Jun 2021 – Aug 2021*

- Translated and refactored legacy CFEngine configuration management code into SaltStack, eliminating major technical debt
- Designed and implemented parallelization for open-source disk usage tool (DUC) in C, achieving ~3x speedups on large file systems

IPVM | Full-Stack Engineer Intern | Ruby on Rails, JavaScript
Bethlehem, PA | *May 2020 – Aug 2020*

- Developed an internal issue management system with due date, assignee, followers, email notifications, sorting/filtering, etc.
- Expanded rich text editor plugin and backend uploader pipeline to support inserting images and files via dialog and drag-and-drop

TEACHING EXPERIENCE

TA for Compiler Design (15-411) | Professor Jan Hoffmann
Pittsburgh, PA | *Spring 2023*

- Transitioned compiler auto-grading infrastructure to Gradescope and helped maintain it for every assignment
- Wrote weekly *recitation* handouts reviewing the various stages of compiler implementation (parsing, typechecking, codegen)
- Led recitation sections, hosted office hours, and graded written assignments every week

TA for Principles of Programming Languages (15-312) | Professors Bob Harper + Jan Hoffmann
Pittsburgh, PA | *Fall 2021, Spring 2022*

- Reworked homework assignments to be more engaging, such as adding a translation task from Python to a statically typed language
- Refactored homework code infrastructure (parsers, typecheckers, interpreters) to be modular and standardized across assignments
- Led recitation sections, hosted office hours, and graded written assignments every week

TA for Principles of Imperative Computation (15-122) | Professors Iliano Cervesato + Dilsun Kaynar
Pittsburgh, PA | *Spring 2020, Fall 2020, Spring 2021*

- Led recitation sections, hosted office hours, and graded written assignments every week

OTHER ACTIVITIES

Competition Programming | Algorithms
High School, Fall 2023

- Identified then coded algorithms and data structures to solve problems under time and space restrictions
- Achieved USA Computing Olympiad Platinum in 2018
- Placed 27th at ICPC NA East Division 2023 (159 teams)

Capture the Flag Cybersecurity Competitions (Team Applicative) | Computer Security
Various times throughout Undergrad

- Located and exploited vulnerabilities in challenge websites and binary executables
- Hunted for hidden information in memory dumps and obscure online databases
- Placed 25th in DarkCTF (808 teams) and 11th in BrixelCTF (824 teams)